REMARKS/ARGUMENTS

Claims 1-34 are pending. Claims 1 and 22 have been amended.

Claims 1-34 stand provisionally rejected under the judicially-created doctrine of obviousness-type double patenting over U.S. Patent Application Serial No. 09/717,645 in view of U.S. Patent 5,649,200 to Leblang et al.

Claims 16, 17, 20, and 21 stand rejected under 35 U.S.C. § 102 as being anticipated by U.S. Patent 5,649,200 to Leblang et al.

Claims 1, 3-6, 8-11, 13-14, 18-19, 22, 24-27, 29-32, and 34 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 5,649,200 to Leblang et al. in view of U.S. Patent 5,890,166 to Eisenberg et al.

Claims 2, 7, 12, 15, 23, 28, and 33 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 5,649,200 to Leblang et al. and U.S. Patent 5,890,166 to Eisenberg et al. in view of U.S. Patent 5,862,325 to Reed et al.

The drawings stand objected to as failing to comply with 37 C.F.R. 1.84(p)(5). As noted above, Applicants are submitting herewith replacement drawing sheets which include the numerals omitted from the original drawings.

The disclosure stands objected to as containing informalities. Applicants have amended the specification as noted above to correct these informalities.

Reconsideration of the Office Action of August 13, 2003 is respectfully requested in view of this response.

Amendments to claims 1 and 22

Claims 1 and 22 have been amended to recite that the link set reference field and the object reference field are included in the link content data structure, and removing the reference to setting the end time field. No new matter has been added by this amendment. The amendment finds support in at least FIG. 3, and the text descriptive of those figures (page 11, line 13 through page 14, line 3).

Non-statutory Double Patenting Rejection

Claims 1-34 stand provisionally rejected under the judicially-created doctrine of obviousness-type double patenting as allegedly being unpatentable over U.S. Patent Application Serial No. 09/717,645 in view of U.S. Patent 5,649,200 to Leblang et al. The examiner contends that, although the conflicting claims are not identical, they are not patentably distinct from each other because the conflicting claims represent obvious variations of the invention recited in the claims of the Application Serial No. 09/717,645.

Applicants note that this is a provisional obviousness-type double patenting rejection, and respectfully defer their response to the merits of the rejection until the allegedly conflicting claims in one of the subject cases are allowed.

Claims 16- 21

Claims 16, 17, 20, and 21 stand rejected under 35 U.S.C. § 102 as being anticipated by U.S. Patent 5,649,200 to Leblang et al. Based on this rejection, claims 18 and 19 stand rejected under 35 U.S.C. § 103. Applicants respectfully disagree with, and traverse, the stated grounds for rejection. As described below, the claims recite various features that are neither taught nor suggested by the prior art. Applicants thus submit that all of the claims are novel and non-obvious over the prior art of record.

Below, applicants discuss various claim features that define over the prior art of record.

"a project management system operative to maintain versions of associations between project management objects"

Claim 16 of the application is directed to a computerized system, and calls for "a project management system operative to maintain versions of associations between project management objects"

In other words, claim 16 calls for project management objects, associations between project management objects, and versions of those associations. The versions of those associations are maintained by the project management system element of the claimed computerized system.

Applicants respectfully maintain that the Leblang reference cited by the examiner does not teach this feature. Leblang deals with a versioning system for source code objects and derived objects. See Leblang, column 5, lines 37-41. Only source code objects, derived objects, directory objects, and symbolic link objects are discussed as being versioned in Leblang. See Leblang, column 5, lines 37-41; column 1, line 66 – column 2, line 5. There is no discussion of project management objects generally, or of their versioning, in Leblang.

Additionally, Leblang does not discuss versioning of associations between objects. Leblang does teach versioning for directory objects. However, Leblang specifically contradicts a view that directory objects (in Leblang) could be analogized to an association of project management objects (in claim 16). From Leblang, column 27, lines 32-36, (emphasis added): "*Directory versions are not directly associated with file versions.* A directory version catalogs the name of an element, not any particular version of that element." Thus, versions of an element can not be associated using a directory. While directory versions may contain a catalog of names for elements, they do not associate the elements.

In contrast, in the claimed invention, associations of specific versions of project management objects are versioned. See the specification, page 9, line 21 through page 11, line 3, specifically on page 10, lines 13-20. As described there, a link set is a set of entities (e.g. project management objects) related by an association, and, "[f]or example, a link set relating a bug report to the set of files that are involved in fixing the bug will comprise a particular version of a link set that relates a particular version of the bug report to the particular versions of the files that were modified to fix the bug." The associations are of specific versions of project management objects. Therefore, the directories in Leblang are not analogous to associations in the claimed invention.

Claims 17, 20, and 21 depend from independent claim 16, and therefore are not anticipated by Leblang for the reasons discussed above.

Claims 18 and 19 depend from claim 17, and have been rejected under 35 U.S.C. § 102 as being anticipated by U.S. Patent 5,649,200 to Leblang et al. in view of U.S. Patent 5,890,166 to Eisenberg et al. The logic for this rejection is based on the 35 U.S.C. § 102 rejection of claims 16 and 17, and therefore the remarks and arguments above apply to distinguish these claims from the teachings of the prior art.

Claims 1-15 and 22- 34

Claims 1, 3-6, 8-11, 13-14, 22, 24-27, 29-32, and 34 stand rejected under 35 U.S.C. § 102 as being anticipated by U.S. Patent 5,649,200 to Leblang et al. in view of U.S. Patent 5,890,166 to Eisenberg et al. Claims 2, 7, 12, 15, 23, 28, and 33 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 5,649,200 to Leblang et al. and U.S. Patent 5,890,166 to Eisenberg et al. in view of U.S. Patent 5,862,325 to Reed et al. Applicants respectfully disagree with, and traverse, the stated grounds for rejection. As described below, the claims recite various features that are neither taught nor suggested by the prior art. Applicants thus submit that all of the claims are novel and non-obvious over the prior art of record.

Below, applicants discuss various claim features that define over the prior art of record.

Claim 1

Claim 1 has been amended to more particularly point out that the link set reference field and the object reference field are in the link content data structure. No new matter has been added by this amendment. The amendment finds support in at least FIG. 3, and the text descriptive of those figures (page 11, line 13 through page 14, line 3).

As amended, claim 1 is directed to a computerized method for adding an association of a project management object to a set of associated project management objects, and calls for:

- creating a link content data structure

- setting a link set reference field in the link content data structure to a value that refers to a link set data structure corresponding to the set of associated project management objects,

- setting an object reference field in the link content data structure to refer to the project management object, and

- setting a start time field in the link content data structure to a value representing the current time.

Link Content Data Structure

No link content data structure as claimed is taught or suggested in Leblang.

Link content data structures according to claim 1 contain a reference to a project management object and a reference to a link set. The association consists of all objects referred to by link content structures which refer to the same link set data structure. Thus one or more link content data structures form the set by each referencing the same link set data structure. This is a concept not present in Leblang. In Leblang, one object (the view, the derived object) may contain a list of objects, however there is never a teaching or suggestion that a number of separate data structures form an association by referring to the same data structure.

The examiner has directed our attention to "view" "configuration record" and "derived object" in Leblang as an example of a link content data structure. However, none of these can be said to teach or suggest the link content data structure as claimed.

More particularly, a link content data structure as per claim 1 includes a link set reference field, an object reference field, and a start time field. No such data structure exists in Leblang. Each of "view", "configuration record" and "derived object" will be discussed in order to make clear that none of these concepts teaches or suggests the claimed invention.

Link Content Data Structure - Views in Leblang

The "view" in Leblang selects a set of versions. (Leblang, col. 9, lines 8-38). Each view contains a "config spec" which is a sequence of configuration rules. (Leblang, col. 10, lines 53-56.) The examiner analogizes the view in Leblang to the link content data structure according to the invention. However, in Leblang, "a view is 'open-ended'. New directories or even entire newly-mounted VOBs are automatically incorporated into a view. There is no need to 'add files to a view' explicitly." (Leblang, col. 15, lines 24-30.) Because no new view is created in Leblang each time a project management object is associated with an existing set of associated project management objects, the creation of a link content data structure upon such an addition can not be found in the "view" of Leblang. New associations

in Leblang do not occur explicitly and no change is made to a view in Leblang when new files are added. Thus, Leblang does not teach or suggest the claimed elements.

Link Content Data Structure - Configuration Records in Leblang

The "configuration record" in Leblang is a "bill of materials" which lists versions of each file element used to create a derived object. (Leblang, col. 30, line 60 – col. 31, line 10). It appears that the examiner is analogizing such a bill of materials to association among objects. Respectfully, applicants argue that this analogy does not show how the elements of the claim can be found in Leblang. Leblang describes the creation of this configuration record, but not its modification. In fact, since it is intended to contain information about the derivation of a specific object, which does not change over time, it would not be reasonable to infer that any change in the configuration record is intended in Leblang.

Thus, there is no way to find a teaching or suggestion in the configuration record of a new association of a project management object by the creation of a data structure including a reference to the project management object (which is referred to in one object reference field of the data structure) and a reference to an existing association of project management objects (referred to in a link set reference field of the data structure). Because of this, it can be seen that configuration records in Leblang do not teach or suggest the elements of claim 1.

Link Content Data Structure - Derived Objects in Leblang

The "derived object" in Leblang is "typically created by running a system build process on particular versions of source objects." (Leblang, col. 5, lines 40-41, col. 2, lines 44-45). The examiner first analogizes Leblang's "derived object" to the link content data structure according to the invention, and then to the link set data structure according to the invention.

Neither analogy can hold. If the derived object of Leblang is to be analogized to the link content data structure, then it must be created upon the addition of a new object to an existing association. No such teaching can be found in Leblang. The derived object is created upon a build.

If the derived object of Leblang is to be analogized to the link set data structure, then upon the addition of a file to the association represented by a derived object, a new data

structure (analogized to the link content data structure) is created, which contains a reference to the link set data structure. But no addition of a new associated file is contemplated by Leblang, nor is any data structure upon such an addition contemplated. Leblang does not show or teach any element of claim 1 by its discussion of derived objects.

In a more general sense, derived objects are created by using a set of objects. For example, an executable may be a derived object. A derived object does not change after its creation, and thus neither does the set of objects it is derived from. Thus, on a general level, any analogy to claim 1, which includes the concept of adding an object, can not be made.

"setting an object reference field to refer to the project management object"

Additionally, the object reference field in a link content data structure is not taught by Leblang. The object references discussed by the examiner ("full pathname" and "a_slink") are, applicants respectfully submit, not contained in any data structures which could be analogized to the link content data structure. The "full pathname" is not disclosed as being stored anywhere in Leblang, but is simply a way to reference to an object, not a field which is set to refer to an object. (Leblang, col. 7, lines 49-55). The "a_slink" is shown as being stored and referring to an object. However, it is not stored in a data structure which includes a reference field which refers to a link set data structure corresponding to a set of associated objects to which the a_slink referenced object is being added. (Leblang, col. 27, lines 4-13).

Thus, this element of claim 1 is also neither taught nor suggested by Leblang.

Generally, the elements of claim 1 are not taught or suggested by Leblang. Additionally, the elements discussed are not found in Eisenberg. Thus, the section 103 rejection of claim 1 should be withdrawn.

Claims 2-5

Dependent claims 2-5 are dependent from and incorporate the elements of claim 1. These features are not taught or suggested by any of the prior art cited by the examiner. Thus, the section 103 rejection of claims 2-5 should be withdrawn.

With reference specifically to claim 5, the development issue data, bug data, project milestone and software specification files of claim 5 are no taught or suggested by Leblang.

The "include header files or object code, meta-data and program source, release notes and scripts" cited by the examiner are different from these project management objects and can not be said to teach or suggest them.

Claims 6-10

Claim 6 (and dependent claims 7-10) is directed to a computerized method for removing an association of a project management object to a set of associated project management objects, and calls for:

- receiving an identifier for a link set corresponding to the set of associated project management objects;

- receiving a reference to the project management object to be removed;

- locating a link content data structure containing the reference to the project management object; and

- setting an end time filed in the link content data structure to a value representing the current time.

The identifier for a link set corresponding to the set of associated objects is not taught or suggested in Leblang. The full pathname and UNIX hard link that the examiner suggests teaches this identifier each refer to a specific object, not to a set of associated objects.

The reference to the project management object contained in a link content data structure is also not taught or suggested in Leblang. The examiner discusses various parts of a derived object (record 514, config rec, link 530) as containing references to the project management object. However, as previously stated, derived objects in Leblang are created by using a set of objects. For example, an executable may be a derived object created from source code files. (Leblang, col. 29, lines 49-57). A derived object does not change after its creation, and thus neither does the set of objects it is derived from. Thus, on a general level, any analogy to claim 6, which includes the concept of adding an object, can not be made.

Because of this static nature of the derived object, the end time as described in Eisenberg can not be combined with the derived object of Leblang. The stated aims of enabling better understanding of real-world interdependencies of versioned objects, their most current versions, and the dynamic state of their being updated by more than one operator over time are not relevant for a data structure which is not versioned or updated.

Therefore, the elements of claim 6 are not taught or suggested by Leblang in view of Eisenberg. Thus, the section 103 rejection of claims 6-10 should be withdrawn. Additionally, with respect to claim 10, please see the above remarks concerning claim 5.

Claims 11-15

Claim 11 (and dependent claims 12-13) is directed to a computerized method for retrieving a set of project management objects associated with a source program management object, including:

- receiving a reference to the source program management object and a time value;
- querying a set of link content data structures to create a set of valid link content data structures, wherein each valid link content data structure contains the reference to the source program management object and further contains a start time value less than or equal to the time value and an end time value that is greater than or equal to the time value;
- creating a set of source link set references comprising the link set reference contained in the set of valid link content data structures; and
- for each source link set reference in the set of source link set references querying the set of link content data structures to create a set of matching project management objects, wherein each matching project management object has a link set reference equal to the source link set reference and further has a start time value less than or equal to the time value and an end time value that is greater than or equal to the time value.

In other words, a reference to a source program management object (X) and a time value (T) are received. A set of link content data structures are queried to find valid link content data structures (VLCDS) which (a) contain a reference to the source program management object X, and (b) contain a start time value less than or equal to the time value T and an end time value that is greater than or equal to the time value T (those which are valid for the time value T). These valid link content data structures VLCDS contain source link set references. A set L of all source link set data structures referred to in the valid link content data structures VLCDS is compiled. A set of all link content items that reference one of the source link sets in L and are valid for the time value T is compiled.

Applicants respectfully maintain that these elements are not found in Leblang. The examiner points to the version selection in Leblang, and analogizes the derived object to the

source program management object. However, the applicants respectfully disagree with examiner's assertion that both the step of creating a set of valid link content data structures *and* the step of creating a set of matching project management objects are "equivalent" to, in Leblang, finding a set of derived objects for reuse. The set of valid link content data structures *and* the set of project management objects (in the invention) are separate elements, and cannot both be analogized to a set of derived objects.

Additionally, with respect to the step of querying a set of link content data structures to create a set of valid link content data structures, the examiner asserts that "getting a reference to a derived object and version-selector to select version of object" is equivalent to finding link content data structures with the time value. However, this is not how Leblang teaches that derived objects for reuse are found. According to Leblang, column 31, lines 52-56: "This determination is made by backmatching the source objects in the configuration record against the current object selection rules to see if the resulting source object version that would be selected for the current build is the same version as that listed in the configuration record." All source objects in each derived object are checked against the current object selection rules, in Leblang. In Leblang, derived objects are therefore not selected for reuse by "getting a reference to a derived object and version-selector."

In claim 11, in this step, link content data structures are found which match in one reference (the reference to the source program management object) and which meet the time value criterion. These valid link content data structures are used to find appropriate source link set references, which are used to find project management objects. This is neither taught nor suggested in Leblang. Thus, the section 103 rejection of claims 11-13 should be withdrawn. Additionally, with respect to claim 13, please see the above remarks concerning claim 5.

In claim 14, when discussing the use of time fields (the claimed second and third fields) to define a range of time that the project management object is associated with the set of associated project management objects, the examiner refers again to the argument in the rejection of claim 11. In this case, as well, applicants respectfully maintain that the analogy of a link content data structure (the data structure claimed in claim 14) to a derived object is flawed. The derived object contains a list of objects which are all related by dint of having been used to create the derived object. No time limit or versioning of this association exists

in a derived object. Leblang does not teach or suggest this limitation of claim 14, and the section 103 rejection of claims 14-15 should be withdrawn.

Claims 22-34 have been rejected by the examiner by reference to the rejections of claims 1-6 and 11, and the Applicants reiterate the arguments presented with reference to those claims above, and maintain that the section 103 rejection of claims 22-34 should be withdrawn.

For all of the foregoing reasons, applicants respectfully submit that this case is now in condition for allowance, and an early notice of allowance is earnestly solicited.

Date: January 13, 2004

Sharon Fenick
Registration No. 45,269

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439